L #6

Call is used to call subroutine (function)

① LCALL : long call

↳ It is 3 byte instruction

↳ first byte is op code

↳ second and third bytes are called Target address

Target address = 16 bits = 2 bytes

$2^{16} = 2^6 \times 2^{10} = 64KB$

LCALL can call subroutine located up to 64 KB

② ACALL : Absolute call

↳ It is 2 byte instruction

↳ 5 bits → opcode

↳ 11 bits is used as target address

Target address = 11 bit → $2^{11} = \underline{2KB}$

- up on reset (power on)

PC = 0000 H

address

```
ORG  0000H
  :
LCALL  SQR1
PC → 0150H INC  A
PCH  PCL   :
  :
ACALL  SQR2
  :
LCALL  SQR1
MOV  R2,A
  :
  :
SJMP $

ORG 250 H
SQR1: MOV  A,R1
      ADD  A,R2
      RET

SQR2: MOV  B,A
  :
  :
      RET

END
```

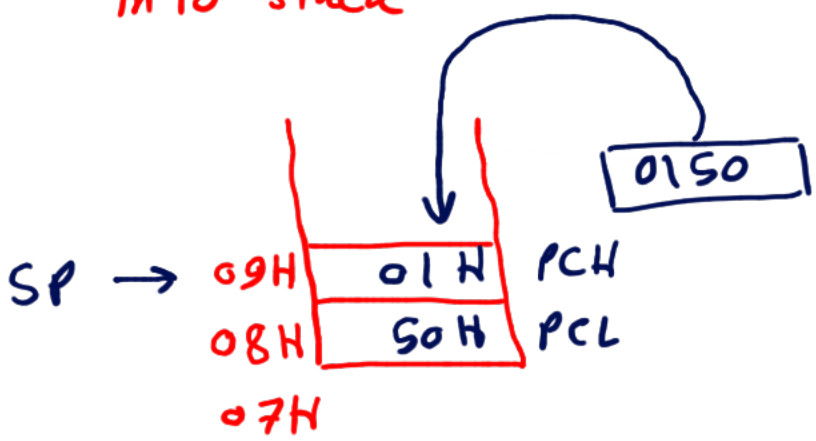
Main program

subroutine

subroutine

* LCALL

① PUSH the content of PC into stack



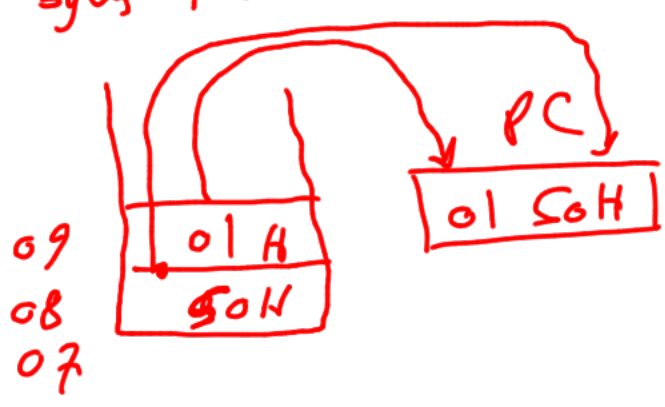
② Load the PC with target address of subroutine

PC = 0250H

---

* RET : return to the main program

pop two bytes from stack into PC



- LCALL increment SP by 2
- RET decrement SP by 2

Show me Stack

address

0150H LCALL SQR1

INC A

⋮

ORG 250H

SQR1 : MOV R2, #80H

PUSH 2

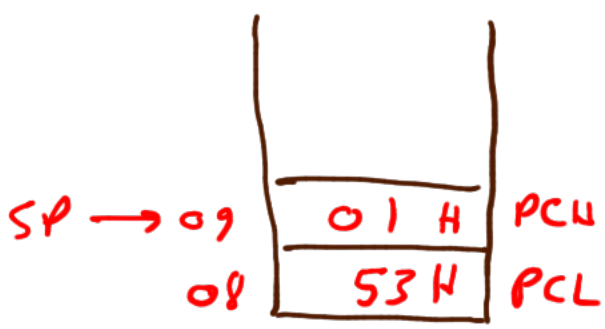
⋮

POP 2

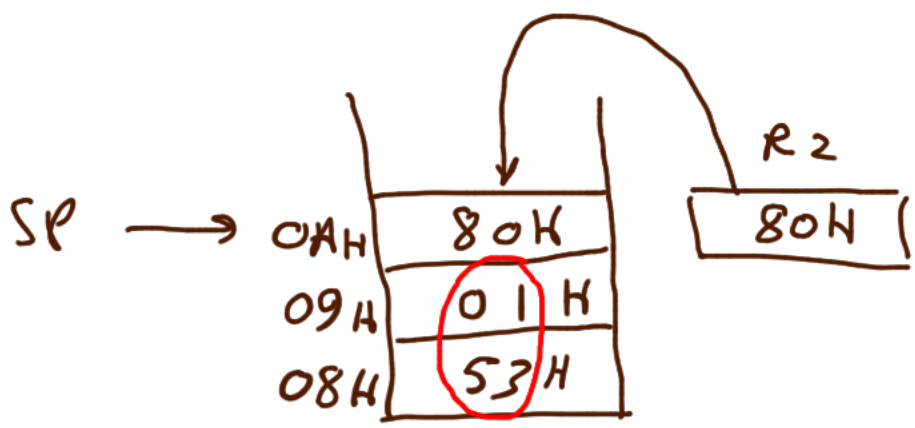
RET

بدونها

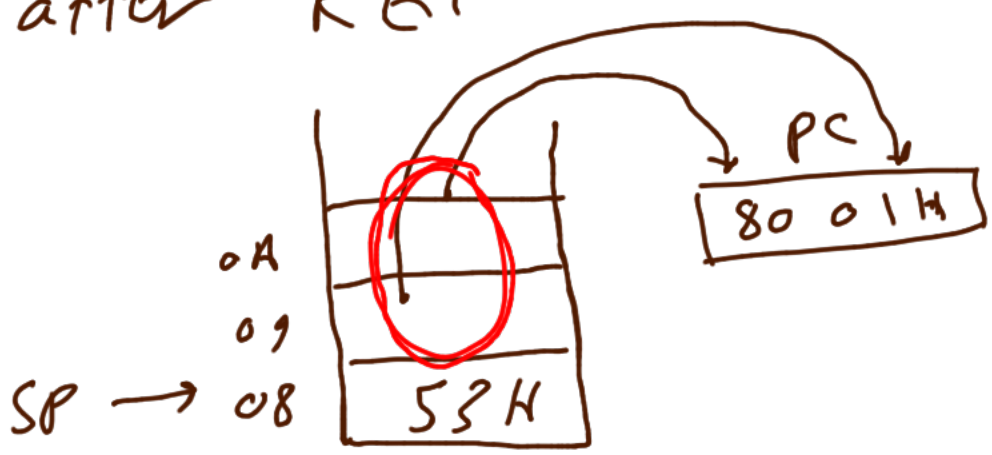
X



after PUSH 2



after RET



- In any subroutine # of push = # of pop instructions, ⑦

---

affects stack

PUSH, POP, RET, LCALL, MOV SP, # ----